A quantum query algorithm for the graph collision problem

Dmitry Gavinsky
NEC Laboratories America, Inc.

Tsuyoshi Ito
NEC Laboratories America, Inc.

Abstract

We construct a new quantum algorithm for the graph collision problem; that is, the problem of deciding whether the set of marked vertices contains a pair of adjacent vertices in a known graph G. The query complexity of our algorithm is $O(\sqrt{n} + \sqrt{\alpha^*(G)})$, where n is the number of vertices and $\alpha^*(G)$ is the maximum total degree of the vertices in an independent set of G. Notably, if G is a random graph where every edge is present with a fixed probability independently of other edges, then our algorithm requires $O(\sqrt{n \log n})$ queries on most graphs, which is optimal up to the $\sqrt{\log n}$ factor on most graphs.

1 Introduction

The quantum query complexity of a function is a natural counterpart of its classical query complexity; namely, it is the number of quantum oracle calls that an algorithm has to make in order to compute its value. Grover's search algorithm [Gro96] gave the first example of a function whose quantum query complexity is significantly smaller than classical: computing the value of the OR function $\bigvee_{i=1}^{n} x_i$ requires $\Omega(n)$ classical queries, but only $O(\sqrt{n})$ quantum queries.

It has been shown by Bennett et al. [BBBV97] that Grover's algorithm is optimal for computing the OR function. Later, Beals et al. [BBC$^+$01] proved that no total function can have the gap between its quantum and classical query complexities larger than polynomial.

Another important problem where a quantum query algorithm can be much faster than a classical one is the element distinctness problem, where n elements are "colored" by an oracle and the goal is to decide whether there are at least two elements of the same color. In 2001 Buhrman et al. [BDH$^+$05] constructed an algorithm that required $O(n^{3/4})$ quantum queries, later a lower bound of $\Omega(n^{2/3})$ was shown by Aaronson and Shi [AS04]. Finally, in 2003 Ambainis [Amb07] gave a new algorithm that had query complexity $O(n^{2/3})$, thus matching the lower bound.

The graph collision problem was first considered by Magniez et al. [MSS07], where it was shown to have quantum query complexity $O(n^{2/3})$. The algorithm used in [MSS07] can be viewed as a natural adaptation of Ambainis' algorithm for the element distinctness problem. On the other hand, the lower bound techniques used in [AS04] don't seem to be applicable to the graph collision problem, and the actual quantum query complexity of it is still an open question.

1.1 Our results and techniques

We present a new quantum algorithm for the graph collision problem. The complexity of our algorithm depends on the properties of the given graph G. Throughout the paper, the quantum query complexity of

a decision problem refers to that with a constant two-sided error.

Theorem 1. *For a graph G on n vertices, the quantum query complexity of the graph collision problem on G is $O(\sqrt{n} + \sqrt{\alpha^*(G)})$, where $\alpha^*(G)$ is the maximum total degree of the vertices in an independent set of G.*

Notably, this implies that the graph collision problem requires only $\tilde{O}(\sqrt{n})$ quantum queries for most graphs in the following sense. Let $\mu_{n,p}$ be the distribution corresponding to choosing a graph on n vertices, where every edge is present with probability p independently of other edges.

Corollary 2. *For arbitrary function $p\colon \mathbb{N} \to [0,1]$, let $G \sim \mu_{n,p(n)}$. Then the (worst-case) quantum query complexity of the graph collision problem on G is almost always*[1] $O(\sqrt{n \log n})$.

The above result is optimal up to the $\sqrt{\log n}$ factor for most random graphs, as computing the OR of n variables can be reduced to solving the graph collision problem on any graph G that contains $\Omega(n)$ non-isolated vertices.

Our algorithm for Theorem 1 works as follows. As a preprocessing, we estimate the sum of the degrees of the vertices in S. If this sum is much larger than $\max\{\alpha^*(G), n\}$, then we answer "S is not an independent set" and halt. This requires $O(\sqrt{n})$ queries, due to the approximate counting algorithm by Brassard, Høyer, and Tapp [BHT98]. To handle the remaining (main) case, we construct a span program with witness size $O(\sqrt{n} + \sqrt{\alpha^*(G)})$. It was shown by Reichardt [Rei09, Rei11] that the quantum query complexity of a promise decision problem is at most a constant factor away from the witness size of a span program computing it.

1.2 Related work

Magniez, Santha, and Szegedy [MSS07] introduced the graph collision problem and gave a quantum algorithm with $O(n^{2/3})$ queries. They used it as a subroutine used in their $O(n^{13/10})$-query algorithm for the triangle finding problem. This $O(n^{2/3})$ is the best known upper bound on the quantum query complexity of the graph collision problem. The best known lower bound for the graph collision problem is $\Omega(\sqrt{n})$, which follows easily from the lower bound for the search problem [BBBV97]. Jeffery, Kothari, and Magniez [JKM12] recently gave a quantum algorithm for the graph collision problem on a bipartite graph which is useful when the given bipartite graph is close to the complete bipartite graph: the query complexity of their algorithm is $\tilde{O}(\sqrt{n} + \sqrt{m})$, where m is the number of missing edges compared to the complete bipartite graph.

Improving the query complexity of the graph collision problem has important consequences. First, improving it is likely to give a better algorithm for the triangle finding problem by applying the same technique as the one used in Ref. [MSS07]. Second, the graph collision problem is equivalent to the evaluation of a 2-DNF formula, and the techniques used in the graph collision problem may be also applicable to the more general k-DNF evaluation.

Our algorithm for the main case of the graph collision problem, including its use of span programs, is inspired by the recent result by Belovs and Lee [BL11].

[1] Cf. Theorem 9 for the corresponding quantitative statement.

2 Preliminaries

We will consider the quantum query complexity of the following problem.

Definition 1 (Graph collision problem)**.** Let $G = (V, E)$ be a graph. The *graph collision problem on* G asks, given oracle access to a string $x \in \{0,1\}^V$, whether there exists an edge $(i, j) \in E$ such that $x_i = x_j = 1$.

Note that graph G is given explicitly to the algorithm, and the only part of the input which needs to be queried is the string $x \in \{0,1\}^V$. We call a vertex i *marked* if $x_i = 1$. Note that the graph collision problem is equivalent to deciding whether the marked vertices form an independent set in G, with the answers "yes" and "no" swapped.

In the rest of the paper, we let $V = [n]$, where $[n]$ denotes the set $\{1, \ldots, n\}$. For a graph $G = (V, E)$ and a set $S \subseteq V$ of vertices, we denote by $\deg(S)$ the sum of degrees of vertices in S. For any graph G, we denote by $\alpha^*(G)$ the maximum total degree of the vertices in an independent set of G; that is,

$$\alpha^*(G) = \max\{\deg(S) \colon S \text{ is an independent set in } G\}.$$

We will use the following form of Chernoff bound, as stated by Drukh and Mansour [DM05].

Lemma 3 (Chernoff bound)**.** *Let $X_1, \ldots, X_n$ be mutually independent random variables taking values in $[0, 1]$, such that $\mathbf{E}[X_i] = \mu$ for all $i \in [n]$. Then for any $\lambda > 1$,*

$$\mathbf{Pr}\left[\sum_{i\in[n]} X_i \geq \lambda n\mu\right] \leq \exp\left(-\frac{n(\lambda-1)^2\mu}{\lambda+1}\right) \leq \exp\bigl((3-\lambda)n\mu\bigr).$$

All logarithms in this paper are natural.

2.1 Span programs

Span program is a linear-algebraic model of computation introduced by Karchmer and Wigderson [KW93] to study the computational power of counting in branching programs and space-bounded computation. In our context, the relevant complexity measure is its *witness size* introduced by Reichardt and Špalek [RŠ08]. We use a formulation closer to that used by Reichardt [Rei09].

Definition 2 (Span program)**.** A *span program* $P = (\mathcal{H}, |t\rangle; V_{10}, V_{11}, \ldots, V_{n0}, V_{n1})$ with n-bit input is defined by a finite-dimensional Hilbert space $\mathcal{H}$ over $\mathbb{C}$, a vector $|t\rangle \in \mathcal{H}$, and a finite set $V_{jb} \subseteq \mathcal{H}$ for each $j \in [n]$ and each $b \in \{0,1\}$. This span program is said to *compute* a function $f \colon D \to \{0,1\}$, where $D \subseteq \{0,1\}^n$, when for $x \in D$, we have $f(x) = 1$ if and only if $|t\rangle$ lies in the subspace of $\mathcal{H}$ spanned by $\bigcup_{j\in[n]} V_{jx_j}$. The vector $|t\rangle$ is called the *target vector* of this span program P.

Definition 3 (Witness size of a span program)**.** Let P be a span program as in Definition 2.

- For an input $x \in f^{-1}(1)$, a *witness* for x is an n-tuple of mappings $w_1, \ldots, w_n$, where $w_j \colon V_{jx_j} \to \mathbb{C}$, such that $|t\rangle = \sum_{j\in[n]} \sum_{|v\rangle\in V_{jx_j}} w_j(|v\rangle)|v\rangle$. The *witness size* on input $x \in f^{-1}(1)$, denoted by $\mathrm{wsize}(P, x)$, is defined as

$$\mathrm{wsize}(P, x) = \min_{(w_1,\ldots,w_n):\text{witness for } x \text{ in } P} \sum_{j\in[n]} \sum_{|v\rangle\in V_{jx_j}} |w_j(|v\rangle)|^2.$$

- For an input $x \in f^{-1}(0)$, a *witness* for x is a vector $|w'\rangle \in \mathcal{H}$ such that $\langle t|w'\rangle = 1$ and $\langle v|w'\rangle = 0$ for every $|v\rangle \in \bigcup_{j\in[n]} V_{jx_j}$. This time, the *witness size* on input $x \in f^{-1}(0)$, again denoted by $\mathrm{wsize}(P,x)$, is defined as

$$\mathrm{wsize}(P,x) = \min_{|w'\rangle:\text{witness for } x \text{ in } P} \sum_{j\in[n],b\in\{0,1\}} \sum_{|v\rangle\in V_{jb}} |\langle v|w'\rangle|^2.$$

- The *witness size* of this span program is $\mathrm{wsize}(P) = \max_{x\in D} \mathrm{wsize}(P,x)$.
- Finally, we denote by $\mathrm{wsize}(f)$ the minimum witness size of a span program which computes f.

For a function $f\colon D \to \{0,1\}$, where $D \subseteq \{0,1\}^n$, we denote by $Q(f)$ the quantum query complexity of f with two-sided error probability at most $1/3$. As is well known, changing the error probability to other constants less than $1/2$ affects the query complexity only within a constant factor.

Theorem 4 (Reichardt [Rei09, Rei11])**.** *Let $f\colon D \to \{0,1\}$, where $D \subseteq \{0,1\}^n$. Then $Q(f)$ and $\mathrm{wsize}(f)$ coincide up to a constant factor. That is, there exists a constant $c > 1$ which does not depend on n or f such that $(1/c)\,\mathrm{wsize}(f) \le Q(f) \le c \cdot \mathrm{wsize}(f)$.*

Proof. Ref. [Rei09] showed that $\mathrm{wsize}(f)$ is equal to the general adversary bound for f, and Ref. [Rei11] showed that the general adversary bound for f and $Q(f)$ coincide up to a constant factor. □

2.2 Quantum algorithm for approximate counting

To detect the case where marked vertices have too many edges, we will use the following result by Brassard, Høyer, and Tapp [BHT98, Theorem 5].

Theorem 5 (Approximate counting [BHT98])**.** *There exists a quantum algorithm which, given integers $N \ge 1$ (domain size) and $P \ge 4$ (precision) and oracle access to a function $F\colon [N] \to \{0,1\}$ satisfying $t = |F^{-1}(1)| \le N/2$, makes P queries to the oracle and outputs an integer $\tilde{t}$, such that*

$$|t - \tilde{t}| < \frac{2\pi}{P}\sqrt{tN} + \frac{\pi^2}{P^2}N$$

with probability at least $8/\pi^2$.

We can remove the assumption that $t \le N/2$ by doubling the size of the domain, and we can reduce the error probability to an arbitrarily small constant by repeating the algorithm constantly many times and taking the majority vote:

Corollary 6. *Let $\varepsilon > 0$ be a constant. Then there exists a quantum algorithm which, given integers $N \ge 1$ (domain size) and $P \ge 4$ (precision) and oracle access to a function $F\colon [N] \to \{0,1\}$, makes $O(P)$ queries to the oracle and outputs an integer $\tilde{t}$ satisfying the following. Let $t = |F^{-1}(1)|$. Then it holds that*

$$|t - \tilde{t}| < \frac{2\sqrt{2}\,\pi}{P}\sqrt{tN} + \frac{2\pi^2}{P^2}N$$

with probability at least $1 - \varepsilon$. (The constant factor hidden in the O-notation of the number of queries depends only on ε and not on N, P, or F.)

3 Algorithm for the main case

The following lemma is useful in the case where not too many edges are incident to marked vertices. In the next section, we will use it with $k = 2\max\{n, \alpha^*(G)\}$ to prove Theorem 1.

Lemma 7. *Let G be a graph on n vertices, and let $k \in \mathbb{N}$. Consider the special case of the graph collision problem on G where it is promised that the set S of marked vertices satisfies $\deg(S) \leq k$.*

(i) There exists a span program for this promise problem whose witness size is at most $\sqrt{2(n+k)}$.

(ii) There exists a quantum algorithm for this promise problem with two-sided error probability at most $1/6$ whose query complexity is $O(\sqrt{n} + \sqrt{k})$.

Proof. Item (ii) follows immediately from item (i) and Theorem 4. In the rest of the proof, we will prove item (i) by constructing a span program explicitly.

Let $\mathcal{H} = \mathbb{C}^{\{0,1\}^n}$, and let

$$|t\rangle = \gamma \sum_{z \in \{0,1\}^n} |z\rangle, \qquad \gamma = \left(\frac{n+k}{2}\right)^{1/4}.$$

For $j \in [n]$ and $b \in \{0,1\}$, let $|s_{jb}\rangle = \sum_{z \in \{0,1\}^n : z_j = b} |z\rangle$. Let $V_{j0} = \varnothing$ and $V_{j1} = \{|s_{j0}\rangle\} \cup \{|s_{i1}\rangle : i \in \mathrm{N}(j)\}$, where $\mathrm{N}(j)$ is the set of neighbors of vertex j in graph G. Define a span program P as $P = (\mathcal{H}, |t\rangle; V_{10}, V_{11}, \ldots, V_{n0}, V_{n1})$.

It is easy to see that P computes the promise problem stated in the lemma. Indeed, if $x \in f^{-1}(1)$, then there exists an edge $ij \in E$ such that $x_i = x_j = 1$. Therefore, $|s_{j0}\rangle \in V_{j1}$ and $|s_{j1}\rangle \in V_{i1}$, which implies that $|t\rangle = \gamma|s_{j0}\rangle + \gamma|s_{j1}\rangle \in \mathrm{span}(V_{j1} \cup V_{i1})$. On the other hand, if $x \in f^{-1}(0)$, then $|w'\rangle = |x\rangle/\gamma$ is a witness for x. Indeed, $\langle t|w'\rangle = 1$, $\langle s_{j0}|w'\rangle = 0$ if $x_j = 1$, and $\langle s_{i1}|w'\rangle = 0$ if $x_j = 1$ and $i \in \mathrm{N}(j)$.

From these witnesses, the witness size of P can be bounded easily. If $x \in f^{-1}(1)$, then the witness stated above shows that the witness size for x is at most $2\gamma^2 = \sqrt{2(n+k)}$. If $x \in f^{-1}(0)$, then the witness stated above shows that the witness size for x is at most $(n+k)/\gamma^2 = \sqrt{2(n+k)}$. □

4 Preprocessing and overall algorithm

In this section, we will prove Theorem 1.

Consider the following quantum algorithm.

1. Compute $\alpha^*(G)$. Let $s = \max\{\alpha^*(G), n\}$. (Because this step does not use any queries to x, how $\alpha^*(G)$ is computed does not matter as long as the query complexity is concerned.)

2. (Preprocessing.) Estimate the number t of pairs $(i,j) \in [n]^2$ such that $ij \in E$ and $x_i = 1$ by running the approximate counting algorithm in Corollary 6 with error probability $\varepsilon = 1/6$ and precision parameter $P = \max\{4, \lceil 7\pi\sqrt{n} \rceil\}$. If the result $\tilde{t}$ of counting satisfies $\tilde{t} > 3s/2$, then answer "yes" and halt. Otherwise, proceed to the next step.

3. (Main case.) Run the algorithm in Lemma 7 (ii) using error probability $1/6$ and parameter $k = 2s$, and answer "yes" or "no" accordingly.

Query complexity. Step 1 does not make any queries to x. Step 2 makes $O(P) = O(\sqrt{n})$ queries. Step 3 makes $O(\sqrt{n}+\sqrt{2s}) = O(\sqrt{n}+\sqrt{\alpha^*(G)})$ queries. Therefore, the query complexity of the whole algorithm is $O(\sqrt{n}+\sqrt{\alpha^*(G)})$.

Correctness. In the rest of this section, we will show that this algorithm reports an incorrect answer with probability at most $1/3$ by considering the following three cases: (a) the correct answer is "yes" and $t \leq 2s$, (b) the correct answer is "yes" and $t > 2s$, (c) the correct answer is "no."

If the correct answer is "yes" and $t \leq 2s$, the only step where the algorithm reports an incorrect answer is step 3, and the promise of the algorithm in Lemma 7 (ii) is satisfied. Therefore, the error probability is at most $1/6$.

If the correct answer is "yes" and $t > 2s$, then with probability at least $5/6$, $\tilde{t}$ satisfies that

$$\begin{aligned}
\tilde{t} &\geq t - |t - \tilde{t}| \\
&> t - \frac{2\sqrt{2}\,\pi}{P}\sqrt{tn^2} - \frac{2\pi^2}{P^2}n^2 \\
&\geq t - \frac{2\sqrt{2}}{7}\sqrt{tn} - \frac{2}{49}n \\
&\geq \sqrt{t}\Big(\sqrt{t} - \frac{2\sqrt{2}}{7}\sqrt{n}\Big) - \frac{2}{49}n \\
&\geq \sqrt{2s}\Big(\sqrt{2s} - \frac{2\sqrt{2}}{7}\sqrt{s}\Big) - \frac{2}{49}s > \frac{3}{2}s.
\end{aligned}$$

Therefore, the algorithm reports "yes" in step 2 alone with probability at least $5/6$. In this case, the promise of the algorithm in Lemma 7 (ii) is not satisfied, but it does not matter what step 3 reports.

Finally, consider the case where the correct answer is "no." In this case, both steps 2 and 3 can report an incorrect answer, and we will bound each of these probability by $1/6$. The correct answer being "no" means that the set of marked vertices is an independent set of G, and therefore it holds that $t \leq \alpha^*(G) \leq s$ by the definitions of $\alpha^*(G)$ and s. This implies that with probability at least $5/6$, $\tilde{t}$ satisfies that

$$\begin{aligned}
\tilde{t} &\leq t + |t - \tilde{t}| \\
&< t + \frac{2\sqrt{2}\,\pi}{P}\sqrt{tn^2} + \frac{2\pi^2}{P^2}n^2 \\
&\leq t + \frac{2\sqrt{2}}{7}\sqrt{tn} + \frac{2}{49}n \\
&\leq s + \frac{2\sqrt{2}}{7}\sqrt{s \cdot s} + \frac{2}{49}s < \frac{3}{2}s.
\end{aligned}$$

Therefore, step 2 reports in an incorrect answer with probability at most $1/6$. Moreover, because the promise of the algorithm in Lemma 7 (ii) is satisfied, step 3 reports an incorrect answer with probability at most $1/6$. By union bound, the overall error probability is at most $1/6 + 1/6 = 1/3$.

5 The case of random graphs

In this section we analyze the query complexity of the graph collision problem defined over random graphs. Recall that we denote by $\mu_{n,p}$ the distribution of random graphs on n vertices, where every edge

is present with probability p, independently of other edges.

We need the following combinatorial lemma.

Lemma 8. *For arbitrary $p \in (0,1]$ and $t \geq 40n\log n$,*

$$\Pr_{G\sim\mu_{n,p}}[\alpha^*(G) \geq t] \leq n^{-14n} + 2\exp\left(\frac{-t^2}{200n^2p}\right).$$

Proof. Assume $G \sim \mu_{n,p}$. For any $t \in \mathbb{N}$, let Y_t be the expected number of independent sets $S \subseteq [n]$ that satisfy $\deg(S) \geq t$. Clearly,

$$\mathbf{Pr}\left[\alpha^*(G) \geq t\right] \leq \mathbf{E}\left[Y_t\right],$$

and therefore we want an upper bound on $\mathbf{E}\left[Y_t\right]$.

For any $i \geq 2$ it holds that

$$\binom{n}{i} \cdot (1-p)^{\binom{i}{2}} \leq \exp\left(i\log n - \frac{pi^2}{4}\right).$$

Let $x_0 \in [n]$, to be fixed later. Then

$$\begin{aligned}
\mathbf{E}\left[Y_t\right] &\leq \sum_{i=1}^{x_0}\binom{n}{i}\cdot(1-p)^{\binom{i}{2}}\cdot\mathbf{Pr}\left[\deg(S)\geq t \;:\; |S|=i\right] \\
&\quad + \sum_{i=x_0+1}^{n}\binom{n}{i}\cdot(1-p)^{\binom{i}{2}} \\
&\leq \sum_{i=1}^{x_0}\exp(i\log n + 3nip - t) + \sum_{i=x_0+1}^{n}\exp\left(i\log n - \frac{pi^2}{4}\right),
\end{aligned}$$

where the last inequality follows from Lemma 3.

Fix $x_0 = \min\left\{\left\lfloor\frac{t}{5np}\right\rfloor, n\right\}$. Then, noting $t \geq 40n\log n$, it holds that

$$\sum_{i=x_0+1}^{n}\exp\left(i\log n - \frac{pi^2}{4}\right) \leq 2\exp\left(-\frac{px_0^2}{8}\right),$$

and we continue:

$$\begin{aligned}
\mathbf{E}\left[Y_t\right] &\leq \exp(\log x_0 + x_0\log n + 3nx_0p - t) + 2\exp\left(-\frac{px_0^2}{8}\right) \\
&\leq \exp\left(-\frac{7t}{20}\right) + 2\exp\left(-\frac{t^2}{200n^2p}\right).
\end{aligned}$$

The result follows. □

The theorem below and Corollary 2 follow immediately from Theorem 1 and Lemma 8.

Theorem 9. *There exists a universal constant C such that for any $p \in (0,1]$, $n \in \mathbb{N}$ and $t \geq 40n\log n$ the following holds. For $G \sim \mu_{n,p}$, the probability that the (worst-case) quantum query complexity of the graph collision problem on G is greater than $C(\sqrt{n}+\sqrt{t})$ is at most $n^{-14n} + 2\exp\left(\frac{-t^2}{200n^2p}\right)$.*

6 Concluding remarks

We gave a quantum algorithm for the graph collision problem on graph G on n vertices whose query complexity is bounded as $O(\sqrt{n} + \sqrt{\alpha^*(G)})$ in terms of the maximum sum of degrees of the vertices in an independent set of G. We used this to show that the graph collision problem has quantum query complexity $\tilde{O}(\sqrt{n})$ for almost all graphs if a graph is chosen at random so that each edge is present with a fixed probability independently of other edges.

We conclude by stating a few open problems. Clearly improving the algorithm so that its query complexity becomes $\tilde{O}(\sqrt{n})$ for *all* graphs is an important open problem. As another direction, the graph collision problem can be defined also for hypergraphs, and it is used in an algorithm for the subgraph finding problem [MSS07], a natural generalization of the triangle finding problem. Extending the present algorithm to the case of hypergraphs is another open problem.

Acknowledgments

Dmitry Gavinsky is grateful to Ryan O'Donnell, Rocco Servedio, Srikanth Srinivasan and Li-Yang Tan for helpful discussions. The authors acknowledge support by ARO/NSA under grant W911NF-09-1-0569.

References

[Amb07] Andris Ambainis. Quantum walk algorithm for element distinctness. *SIAM Journal on Computing*, 37(1):210–239, 2007. arXiv:quant-ph/0311001v8.

[AS04] Scott Aaronson and Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the ACM*, 51(4):595–605, 2004. arXiv:quant-ph/0112086v1.

[BBBV97] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997. arXiv:quant-ph/9701001v1.

[BBC+01] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001. arXiv:arXiv:quant-ph/9802049v3.

[BDH+05] Harry Buhrman, Christoph Dürr, Mark Heiligman, Peter Høyer, Frédéric Magniez, Miklos Santha, and Ronald de Wolf. Quantum algorithms for element distinctness. *SIAM Journal on Computing*, 34(6):1324–1330, 2005. arXiv:quant-ph/0007016v2.

[BHT98] Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum counting. In *Automata, Languages and Programming: 25th International Colloquium, ICALP'98: Proceedings*, volume 1443 of *Lecture Notes in Computer Science*, pages 820–831, 1998. arXiv:quant-ph/9805082v1.

[BL11] Aleksandrs Belovs and Troy Lee. Quantum algorithm for k-distinctness with prior knowledge on the input. arXiv:1108.3022v1 [quant-ph], 2011.

[DM05] Evgeny Drukh and Yishay Mansour. Concentration bounds for unigram language models. *Journal of Machine Learning Research*, 6(Aug):1231–1264, 2005.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC)*, pages 212–219, 1996. arXiv:quant-ph/9605043v3.

[JKM12] Stacey Jeffery, Robin Kothari, and Frédéric Magniez. Improving quantum query complexity of Boolean matrix multiplication using graph collision. arXiv:1112.5855v2 [quant-ph], 2012.

[KW93] Mauricio Karchmer and Avi Wigderson. On span programs. In *Proceedings of the 8th Annual Structure in Complexity Theory Conference (CoCo)*, pages 102–111, 1993.

[MSS07] Frédéric Magniez, Miklos Santha, and Mario Szegedy. Quantum algorithms for the triangle problem. *SIAM Journal on Computing*, 37(2):413–424, 2007. arXiv:quant-ph/0310134v3.

[Rei09] Ben W. Reichardt. Span programs and quantum query complexity: The general adversary bound is nearly tight for every boolean function. In *50th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 544–551, 2009. arXiv:0904.2759v1 [quant-ph].

[Rei11] Ben W. Reichardt. Reflections for quantum query algorithms. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 560–569, 2011. arXiv:1005.1601v1 [quant-ph].

[RŠ08] Ben W. Reichardt and Robert Špalek. Span-program-based quantum algorithm for evaluating formulas. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 103–112, 2008. arXiv:0710.2630v3 [quant-ph].